

DOCUMENTO INTERNO DE TRABAJO

PROGRAMACIÓN DE PEDIDOS EN UNA MÁQUINA DE LA VIDA REAL CON TIEMPOS DE PREPARACIÓN DEPENDIENTES DE LA SECUENCIA

Autores:

MAYRA D'ARMAS

RAMÓN COMPANYS

Barcelona, octubre de 2005

LABORATORIO DE ORGANIZACIÓN INDUSTRIAL
DEPARTAMENTO DE ORGANIZACIÓN DE EMPRESAS
UNIVERSIDAD POLITÉCNICA DE CATALUÑA

PROGRAMACIÓN DE PEDIDOS EN UNA MÁQUINA DE LA VIDA REAL CON TIEMPOS DE PREPARACIÓN DEPENDIENTES DE LA SECUENCIA

(Octubre, 2005)

**M. D'Armas y R. Companys
LOI-DOE-UPC**

En este trabajo se evalúan algoritmos heurísticos de exploración de entornos (AED, ANED, SA, TS, GA y GRASP) en la programación de pedidos en una máquina de la vida real, con el objetivo de minimizar la suma de retrasos. El caso estudiado se diferencia de los problemas convencionales en que los tiempos de preparación de las operaciones están separados de los tiempos de procesamiento y son dependientes de la secuencia. Los resultados computacionales revelan que la Búsqueda Tabú funciona mejor que los otros algoritmos aplicados.

Keywords: Single machine; Sequence-dependent setup times; Total tardiness; Heuristics

1. INTRODUCCIÓN

El presente trabajo tiene por objetivo evaluar el comportamiento de las heurísticas de exploración de entornos AED, ANED, Recocido Simulado, Algoritmos Genéticos, Búsqueda Tabú y GRASP en la resolución de un problema de programación de operaciones en un ambiente real de una máquina. El problema investigado es el de encontrar una secuencia, en un ambiente de una máquina, con familias de productos, y donde los tiempos de preparación de la máquina dependen de la secuencia de las pedidos de productos a fabricar, de forma tal que se cumplan con los tiempos de entrega, o en su defecto, que minimice la suma de retrasos.

El problema planteado se ha aplicado a una industria real que se dedica a la fabricación de productos de acero, y que tiene como uno de sus objetivos, optimizar la producción y los beneficios de la empresa en función de las exigencias del mercado, en cuanto al volumen, calidad y oportunidad.

La empresa en cuestión produce alambrón, con un total de dieciocho familias de productos diferentes que tienen un diámetro que varía desde 5,5 mm hasta 12,5 mm.

El alambrón es un producto de acero de sección transversal circular y superficie lisa, que se obtiene por laminación en caliente de palanquillas. Es fabricado de aceros al carbono, en varias calidades, cada una de las cuales requiere una combinación específica de prácticas de refinación del acero, laminación de palanquillas,

acondicionado de las mismas, enfriamiento e inspección para cumplir con los requerimientos de los usuarios. Algunos de los usos más frecuentes del alambraón son: la elaboración de clavos, grapas, tornillos y tuercas, resortes, guayas, varillas y alambre para electrodos, entre otros.

Durante el proceso de fabricación, para pasar de una familia de productos a otra, se requiere cambiar los cilindros y anillos laminadores del tren de laminación, lo que origina unos tiempos de preparación que dependen tanto de la familia a ser fabricada como de la inmediatamente precedente. De esta manera, la matriz de tiempos de preparación expresada en horas es la siguiente:

<i>ST_{h,i}</i>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	0	2	2	2	1	2	2	1	1	2	2	2	2	2	2	2	2	2
2	2	0	2	2	2	1	2	2	2	1	1	2	2	2	2	2	2	2
3	2	2	0	1	2	2	1	2	2	2	2	2	2	2	2	2	2	2
4	2	2	1	0	2	2	1	2	2	2	2	2	2	2	2	2	2	2
5	1	2	2	2	0	2	2	1	2	2	2	2	2	2	2	2	2	2
6	2	1	2	2	2	0	2	2	2	1	2	2	2	2	2	2	2	2
7	2	2	1	1	2	2	0	2	2	2	1	2	2	2	2	2	2	2
8	1	2	2	2	1	2	2	0	1	2	2	2	1	2	2	2	2	2
9	1	2	2	2	2	2	2	1	0	2	2	2	1	2	2	2	2	2
10	2	1	2	2	2	1	2	2	2	0	2	2	2	2	1	2	2	2
11	2	1	2	2	2	2	1	2	2	2	0	2	2	2	1	1	2	2
12	2	2	2	2	2	2	2	2	2	2	2	0	2	2	2	2	2	2
13	2	2	2	2	2	2	2	1	1	2	2	2	0	1	2	2	2	2
14	2	2	2	2	2	2	2	2	2	2	2	2	1	0	2	2	2	2
15	2	2	2	2	2	2	2	2	2	1	1	2	2	2	0	1	2	2
16	2	2	2	2	2	2	2	2	2	2	1	2	2	2	1	0	2	1
17	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	1
18	4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1	0

La jornada de trabajo es de tres turnos de ocho horas cada uno, a saber:

- (1) primer turno desde 7:00 a.m. hasta 3:00 p.m.
- (2) segundo turno desde 3:00 p.m. hasta 11:00 p.m.
- (3) tercer turno desde 11:00 p.m. hasta 7:00 a.m. Las fechas de vencimiento, expresadas en horas, se consideran que coinciden con el fin de la tercera jornada de trabajo.

2. TERMINOLOGÍA UTILIZADA

Siguiendo a Conway, Maxwell y Miller (1967), la nomenclatura utilizada consta de cuatro campos A/B/C/D, donde A corresponde al número de piezas (n indica un número de piezas arbitrario). B describe el número de máquinas en el taller (m indica un número arbitrario). C se refiere al tipo de flujo de las piezas por el taller, y generalmente toma uno de los valores F, P, R o G:

F: indica flujo regular (flowshop), todas las piezas tienen esencialmente la misma ruta.

P: indica un caso particular del anterior, donde todas las máquinas tienen la misma secuencia de piezas.

R: indica rutas aleatorias (randomly routed job shop)

G: indica flujo general.

D se refiere al índice de eficiencia elegido para evaluar los programas, entre los que se encuentran c_{\max} (instante de salida de la última pieza del taller), F_{\max} (tiempo de permanencia en el taller de la pieza que permanece más tiempo), F_{med} (tiempo medio de permanencia en el taller), T_{\max} (retraso de la pieza que se retrasa más), T_{med} (retraso medio).

Como ejemplo de la nomenclatura de Conway, Maxwell y Miller, $n/2/F/F_{\max}$ representa la secuencia en un número arbitrario de piezas en un flow-shop de dos máquinas de manera que se minimice el tiempo máximo de permanencia.

El problema investigado en este trabajo es parecido al $n/1/ST$, sin embargo, los mencionados autores no tomaron en cuenta los tiempos de preparación, asumiendo que el tiempo de procesamiento p_i incluye cualquier preparación de la máquina para realizar la operación y(o) cualquier desmontaje requerido después de la operación.

Por lo tanto, siguiendo la notación previamente citada, se agrega al campo C las siglas: $ST_{sd,b}$ = sequence-dependent batch setup times, para describir el problema de una máquina con tiempos de preparación dependientes de la secuencia y familia de productos, transformando esta nomenclatura en $n/1/ST_{sd,b}/\Sigma T$.

La notación usada es la siguiente:

n	número de piezas
i	índice de las piezas; $i \in I = \{1, 2, 3, \dots, n\}$
p_i	tiempo de operación de la pieza i
b_i	familia a la que pertenece la pieza i
ST_{hi}	tiempo de preparación de la pieza i cuando h ha sido la pieza anterior de la secuencia
d_i	instante comprometido de salida de la pieza i (due date)
r_i	instante de entrada de la pieza i en el taller (ready time)
c_i	instante en que la pieza sale del taller (completion time)
w_i	tiempo de espera de la pieza i (waiting time)
L_i	diferencia entre el instante de salida real y el previsto (lateness)
T_i	retraso (tardiness)

3. FORMULACIÓN DEL PROBLEMA

En un ambiente de una máquina, se tiene un conjunto de n pedidos que deben ser secuenciados. Se asume que la máquina puede procesar una sola pieza a la vez y que está disponible en el instante cero. Los pedidos están todos disponibles al inicio del proceso (tiempo cero) y tienen un tiempo d_i límite de finalización o entrega. Se asume que los pedidos están clasificados en varias familias y que el tiempo de preparación ST se produce cuando se pasa de una familia de productos a otra.

Se asume también que hay un tiempo de preparación de la máquina y que éste depende del orden de la secuencia, de tal manera que hay un tiempo de preparación que debe preceder al inicio del procesamiento del pedido dado, que depende de la familia del producto a ser procesado y la familia del producto que lo precede. La matriz de tiempos de preparación no necesariamente tiene que ser simétrica.

Para cada pedido i se conoce el tiempo de operación p_i , el tiempo de entrega comprometido d_i y la familia a la que pertenece dicho pedido b_i . Además, se conocen los tiempos de preparación de la máquina ST al pasar de una familia a otra.

Según la secuencia escogida, para cada pedido i el instante en que el pedido sale del taller c_i , viene dado por: $c_i = r_i + p_i + ST_{hi}$, y su retraso T_i viene dado por: $T_i = \max. \{0, c_i - d_i\}$. El objetivo es encontrar una secuencia de los pedidos que minimice la suma del retraso:

$$[\text{MIN}] Z = \sum_{i=1}^n T_i$$

Minimizar la suma de los retrasos es equivalente a minimizar el retraso medio, ya que el retraso medio de las piezas (T_{med}) viene dado por:

$$\frac{1}{n} \sum_{i=1}^n T_i$$

4. ESTRATEGIA DE SECUENCIACIÓN DE LA SOLUCIÓN INICIAL

El aspecto fundamental de cuatro (Algoritmo Exhaustivo de Descenso, Algoritmo No Exhaustivo de Descenso, Recocido Simulado y Búsqueda Tabú) de los seis procedimientos aplicados en este trabajo, consiste en comenzar desde una solución inicial y generar un conjunto de soluciones vecinas. Según Johnson et al (1989) puede ser ventajoso comenzar con una buena solución más que con una generada aleatoriamente, es decir, por lo general, se obtiene un mejor resultado final si la solución de partida es buena.

En función de esto se han generado cuatro soluciones iniciales diferentes (EDD, SST-EDD, CR1 y CR2), mediante la aplicación de las cuatro heurísticas directas, expuestas a continuación:

4.1 EDD (Earliest Due Date)

Consiste en ordenar los pedidos, en orden creciente, de acuerdo con la fecha de vencimiento. Para cada pedido i se considera la fecha de vencimiento d_i y el tiempo de preparación $ST_{h,i}$ (dependiente de la familia del producto fabricado en el pedido h). Dada las características del problema tratado, esta regla favorece los pedidos más prioritarios, sin embargo no aprovecha la ventaja de reducir los tiempos de preparación al procesar, sucesivamente, pedidos de la misma familia de productos.

4.2 SST-EDD (Shortest Setup Time – Earliest Due Date)

Consiste en ordenar los pedidos por familias de acuerdo con el tiempo de preparación más corto cuando se cambia de una familia a otra (para favorecer un tiempo mínimo de cambios entre familias), y, además, secuenciar los pedidos entre familia por orden creciente de fechas de vencimiento. Para cada pedido i se considera la familia b_i , la fecha de vencimiento d_i y el tiempo de preparación $ST_{h,i}$ (dependiente de la familia del producto fabricado en el pedido anterior h).

4.2 CR1 (Cociente)

Índice crítico para cada pieza i , basado en De Castro et al (2003), que consiste en calcular el índice de prioridad por medio de la fecha de vencimiento dividida por la suma entre el tiempo de preparación de la familia del pedido i (dependiente de la familia del producto fabricado en el pedido anterior h) y el tiempo de procesamiento del pedido i . Los pedidos que tienen el CR1 más pequeños se asignan primero. Para cada pedido i se considera la fecha de vencimiento d_i , el tiempo de preparación $ST_{h,i}$ y el tiempo de procesamiento u operación p_i : $CR1 = d_i / (ST_{h,i} + p_i)$

4.3 CR2 (Suma ponderada)

Basado en De Castro et al (2003), consiste en asignar los pedidos tomando en cuenta el índice de prioridad para cada pedido i , el cual pondera la fecha de vencimiento, el tiempo de preparación de la familia del pedido i (dependiente de la familia del producto fabricado en el pedido anterior h) y el tiempo de procesamiento del pedido: $CR2 = a d_i + (1-a) (p_i + ST_{h,i})$.

Se ha empleado un alfa igual a 0,2 de acuerdo con lo propuesto por Companys y D'Armas (2005).

5. HEURÍSTICAS DE MEJORA

5.1 Algoritmo Exhaustivo de Descenso (AED)

A partir de la solución en curso se generan y evalúan todos los vecinos, si el mejor de ellos (en caso de búsqueda de un mínimo el de menor valor) es mejor que la solución

en curso se toma como nueva solución en curso y se reitera el procedimiento; en caso contrario, si el mejor vecino es peor o igual a la solución en curso el procedimiento se da por terminado.

5.2 Algoritmo No Exhaustivo de Descenso (ANED)

A partir de la solución en curso se generan y evalúan en cierto orden sus vecinos, si uno de ellos es mejor que la solución en curso se toma como nueva solución en curso (sin terminar la generación de los vecinos de la solución primitiva) y se prosigue aplicando el procedimiento a los vecinos de la nueva solución en curso; cuando se han generado todos los vecinos de una determinada solución en curso sin que ninguno sea mejor el procedimiento se da por terminado.

5.3 Recocido Simulado (SA, Simulated Annealing)

La aproximación SA fue desarrollada por Kirkpatrick, Gelatt y Vecchi (1983) y se le conoce como Recocido Simulado dado la analogía entre la simulación del recocido de sólidos y el problema de la resolución de los grandes problemas de optimización combinatoria.

SA parte de una solución inicial determinada por una heurística, y mediante la exploración de su entorno, trata de encontrar la solución óptima. El método consiste de iteraciones, donde se compara la solución en curso con la solución vecina generada aleatoriamente. Si la solución vecina es mejor que la solución en curso, entonces la solución vecina pasa a ser la solución en curso. En el caso de que la solución vecina sea peor que la solución en curso se puede tomar esta solución vecina como la solución en curso, con una probabilidad que depende de la diferencia entre la solución vecina y la solución en curso y la temperatura T .

Para el problema estudiado, la regla de generación de la probabilidad de aceptación de la solución vecina que es peor que la solución en curso sigue la distribución Boltzmann:

$$p(X_{vec}) = e^{-[f(X_{vec}) - f(X_{cur})]/T}$$

Donde la función tomará valores entre [0,1], y dependerá de la solución vecina, la solución en curso y la temperatura T en la iteración n .

Para la implementación del recocido se han tomado los valores siguientes: (a) Temperatura inicial $T = 0,4$; (b) Tasa de disminución de la temperatura o rata de

enfriamiento $r = 0,95$; y (c) Size factor = 16. La generación de los vecinos se ha realizado al azar. Para ello se generan, aleatoriamente, dos piezas a permutar en la secuencia dando lugar a una nueva secuencia vecina. La secuencia vecina se diferencia de la secuencia madre en el orden de realización de dos piezas.

5.4 Búsqueda Tabú (TS, Tabu Search)

TS es un procedimiento metaheurístico de alto nivel introducido y desarrollado en su forma actual por Fred Glover (1989). Su filosofía se basa en la explotación de diversas estrategias inteligentes para la resolución de problemas, basadas en procedimientos de aprendizaje. (Glover y Melián, 2003). A diferencia de otros algoritmos basados en técnicas aleatorias de búsqueda de soluciones cercanas, TS es determinista ya que elimina el azar en sus decisiones y la búsqueda del óptimo está guiada por una estrategia basada en el uso de estructuras de memoria, que guardan soluciones a corto y largo plazo.

El aspecto fundamental del procedimiento consiste en comenzar desde una solución inicial y generar un conjunto de soluciones de su vecindario, de estas elegir la mejor de las vecinas aún cuando ésta sea peor que la solución en curso. La característica importante de TS es esencialmente la construcción de una lista tabú de movimientos: aquellos movimientos que no son permitidos (movimientos tabú) en la presente iteración. La razón de esta lista es la de excluir los movimientos que pueden hacer regresar a algún punto de una iteración anterior, es decir evitar regresar al mismo óptimo local.

Debido a que la lista tabú puede prohibir soluciones que tengan una calidad superior a la de las soluciones ya conocida, se hizo uso del concepto nivel de aspiración. El criterio que se ha tomado en este trabajo es seleccionar aquellas secuencias de operaciones cuyo valor de ΣT , retraso total, sea inferior al de la mejor solución hallada hasta el momento. Primero se comprueba si la solución vecina actual satisface el nivel de aspiración, si es así, la solución vecina sustituye a la solución en curso y a la mejor. En caso contrario, si dicha solución vecina es igual o peor que la mejor, entonces se compara con los atributos de la lista tabú. La generación de vecinas se ha realizado mediante la exploración de todo el entorno de la solución en curso, es decir, dada una secuencia de operaciones se generan todos los vecinos

mediante la permutación entre dos piezas, haciéndose todos los cambios posibles de dos posiciones (que podrán ser contiguas o no).

6. GRASP

Fue desarrollado originalmente por Feo y Resende (1989) al estudiar un problema de cobertura de alta complejidad combinatoria. La técnica empleada para esta metaheurística es de tipo iterativo. Cada iteración en GRASP consta generalmente de dos pasos: la fase de construcción y el procedimiento de búsqueda local. En el primero se construye una solución tentativa, que luego es mejorada mediante un procedimiento de intercambio hasta que se llega a un óptimo local. En la fase constructiva, GRASP toma en cuenta la función objetivo con la intención de que al término de la iteración se cuente con una solución de alta calidad, sobre la cual se efectúa una mejora o Fase II.

En este trabajo se ha aplicado una heurística GRASP basada en Ribas y Companys (2004). En la primera Fase, para disponer de una solución inicial se construye una solución inicial a partir de Índice Crítico CR2, secuenciándose las piezas en orden no decreciente de su índice. Se toma el menor valor del índice CR2, y se determina un valor de referencia mediante un incremento del veinte por ciento (20%) del menor valor. Se elige al azar una de las piezas cuyo índice CR2 sea menor que el valor de referencia, y se asigna como primera pieza a procesar. Se recalculan los índices CR2 y se ordenan nuevamente las piezas, desde la segunda posición, de acuerdo con un orden no decreciente de los índices recalculados.

En la segunda Fase, el post-procesamiento, se ha aplicado un procedimiento de intercambio de 2-intercambio, y una búsqueda a k-vecindades. El procedimiento de mejora aplica una variante de la heurística ANED, donde cada iteración recorre el conjunto de soluciones de forma distinta pudiendo acceder a vecindarios no explorados.

Para lograr esto, se define un nuevo vector de posiciones (*rev*), que permite codificar los punteros de posición que se usan durante la exploración del vecindario, con el fin de evitar recorrer dicho vecindario siempre de la misma forma. Inicialmente, el vector *rev* contiene de forma ordenada las diferentes posiciones que una pieza puede

ocupar en la secuencia. No obstante, posteriormente se mezclan al azar las posiciones y, durante la aplicación del procedimiento de mejora de la heurística ANED, se codifican, a través del vector *rev*, los punteros de posición utilizados para explorar el vecindario de una solución. Es decir, para una posición $k1$, $k2$ se toma su equivalente $k1rev$, $k2rev$ del vector *rev*, siendo $k1rev = rev(k1)$ y $k2rev = rev(k2)$, y se aplica el procedimiento con estos nuevos punteros.

7. ALGORITMO GENÉTICO (GA, Genetic Algorithm)

GA es un algoritmo de búsqueda que explora un espacio de solución que simula procesos en un sistema natural hacia la evolución, específicamente aquellos que siguen el principio de la supervivencia en función de la adaptabilidad. (Díaz et al, 1996). Fue desarrollado por Holland (1975), y se distingue muy claramente de todos los anteriores, básicamente por el hecho de que en cada iteración se tiene un conjunto de soluciones, o población en curso y no una única solución en curso. Las soluciones sucesoras se obtienen a partir de parejas constituidas con los elementos de la población y no mediante la transformación de la solución en curso.

7.1 Población inicial

En este trabajo se ha aplicado una combinación de perturbaciones aleatorias con reglas de secuenciación, que permiten garantizar que la población inicial sea heterogénea y que se inicie con soluciones que tengan una calidad aceptable. Así mismo, la dimensión de la población se ha tomado igual al número de operaciones de tal manera que la población tiene una relación directamente proporcional con la dimensión del problema a tratar.

En cuanto a su composición, la población inicial se ha dividido en dos partes iguales, las cuales se diferencian entre sí por el esquema de confección de los cromosomas que conforman dicha población. La mitad de la población se forma secuenciando todas las operaciones, en orden creciente, de acuerdo con la fecha de vencimiento. Posteriormente, se generan aleatoriamente, dos genes a permutar en la secuencia dando lugar a una nueva secuencia. La otra mitad, se forma secuenciando todas las operaciones de acuerdo con el Índice Crítico CR2 en orden no decreciente de su índice. Posteriormente, se generan aleatoriamente, dos genes a permutar en la

secuencia dando lugar a una nueva secuencia.

7.2 Selección

Mediante la selección se determinan los individuos de la población que tendrán descendencia. En este trabajo el problema estudiado tiene como objetivo minimizar los retrasos en las entregas de los pedidos, por lo tanto la medida de la calidad de un individuo (solución) que se ha considerado es el valor de ST . El proceso de selección se ha realizado mediante la combinación de las mejores secuencias y de forma aleatoria, es decir un 50% de las secuencias a cruzar lo conforman las mejores soluciones de la población actual y el otro 50% se elige aleatoriamente.

7.3 Cruces

El procedimiento de cruce consiste en combinar dos secuencias padres según ciertos criterios. Inicialmente, las soluciones progenitoras del grupo seleccionado, se han emparejado por orden de bondad de la solución, es decir, la primera solución se empareja con la segunda, la tercera con la cuarta, etc. dentro de la lista de soluciones elegidas ordenadas de acuerdo con el valor de ST . A continuación, una vez que se tienen las parejas, se procede a realizar el cruce entre ambas, las cuales generan dos soluciones descendientes.

El procedimiento de cruce que se ha aplicado es el Cruce PMX (Partially-Matched Crossover). Dado dos soluciones progenitoras, se eligen dos puntos de cruce al azar, luego, el procedimiento copia los genes de uno de los padres directamente a las mismas posiciones en el hijo. Las posiciones restantes se llenan con los valores que aún no han sido utilizados en el mismo orden en que se encuentran en uno de los padres.

7.4 Mutaciones

La mutación es un mecanismo que, a imitación de lo que sucede en la naturaleza con los seres vivos, fomenta la diversidad. Para el problema, se ha fijado la probabilidad en un treinta por ciento, de acuerdo con los resultados de Fernández-Baños (2003). Es decir, el procedimiento generará un número aleatorio entre 0 y 1, si este valor es inferior al parámetro de probabilidad fijado (0,3) se realiza la mutación en la secuencia. En el supuesto que lo haga, la mutación consiste en intercambiar un gen, seleccionado al azar, con su consecutivo en la secuencia.

7.5 Regeneración

Una vez incrementado la población con los descendientes, mutados o no, hay que reducirla al tamaño conveniente y este proceso es el que se denomina regeneración. Se trata, por lo tanto, de elegir entre las soluciones actuales, un cierto número de ellas. En este proceso de regeneración se han seleccionado las mejores secuencias de las dos poblaciones de soluciones, progenitoras e hijas. Es decir, la nueva población generada para la siguiente iteración está formada por la mitad del total de las soluciones (padres + hijos).

8. EXPERIENCIA COMPUTACIONAL

En esta sección se describen las pruebas computacionales que se usaron para evaluar la eficiencia de las distintas heurísticas de exploración de entornos. Los algoritmos se codificaron en Visual Basic 6.0 y las pruebas se realizaron en un computador personal Pentium IV de 3,00 GHz y 1,00 GB de Ram

La viabilidad de los algoritmos se ha estudiado en diecisiete ejemplares suministrados por una empresa que fabrica productos de acero, donde el número de pedidos, n , fabricados en cada uno de los ejemplares estudiados varía entre ocho y dieciséis pedidos, con un número de 18 familias de productos diferentes que tienen un diámetro que varía desde 5,5 mm hasta 12,5 mm. Para pasar de una familia de productos a otra, se requiere cambiar los cilindros y anillos laminadores del tren de laminación, lo que origina unos tiempos de preparación que dependen tanto de la familia a ser fabricada como de la inmediatamente precedente.

La información suministrada contempla:

- Los tipos de productos a fabricarse y sus respectivas cantidades requeridas expresadas en toneladas.
- La familia a la que pertenece cada pedido.
- La familia procesada anteriormente (para la cual la máquina está preparada inicialmente)
- La fecha de vencimiento en la que se precisa el pedido (tiempo de entrega de

cada pedido en horas)

- Los tiempos de preparación de la máquina para pasar de una familia de productos a otra expresado en horas
- El tiempo unitario de producción de cada familia de productos expresado en horas/ toneladas y el tiempo de operación de cada pedido (en horas).

Cada ejemplar se ha resuelto empleando cada una de las cuatro estrategias de secuenciación de la solución inicial, y posteriormente se le ha aplicado los procedimientos heurísticos AED, ANED, SA y TS, seleccionándose para cada procedimiento el mejor resultado de las cuatro soluciones mejoradas. Además, se ha resuelto cada ejemplar con las metaheurísticas GRASP y GA.

Los procedimientos heurísticos se han juzgado de acuerdo con Bulfin y Sipper (1998), por su calidad (diferencia entre una solución heurística y la óptima) y eficacia (esfuerzo realizado para obtener la solución).

Para juzgar la eficacia de los algoritmos se ha computado el tiempo promedio de ejecución, de cada uno de los procedimientos, necesario para resolver un ejemplar. Debido a que en el problema estudiado no se tiene la solución óptima, para juzgar la calidad se ha determinado la diferencia entre la solución heurística y la mejor solución, es decir, la solución heurística se ha comparado en términos del porcentaje de error relativo. (Rajendran y Ziegler, 2003), (Allahverdi y Aldowaisan, 2000), (Yang, Kreipl y Pinedo, 2000), (Raghu y Rajendran, 1995). A cada uno de las heurísticas de exploración de entornos, se le ha determinado el porcentaje de error relativo de cada ejemplar, mediante la expresión:

$$[100x (Heurística- Mejor Solución)/Mejor Solución].$$

Posteriormente, se le ha calculado el promedio de todos los ejemplares de la colección.

Los resultados de los retrasos totales, expresados en horas, para cada uno de los ejemplares estudiados se muestran a continuación en la tabla I. Se ha indicado para cada ejemplar el número de pedidos y de familias, señalándose con un asterisco la mejor solución, es decir el menor retraso total obtenido mediante la aplicación de los diferentes procedimientos heurísticos.

Tabla I. Caso real. Retraso total expresado en horas.

Ejemplar	n	b	RETRASO TOTAL (HR)					
			AED	ANED	SA	GRASP	TS	GA
1	16	18	230	259	265	269	17*	274
2	15	18	639	637	647	639	563*	649
3	8	18	192	192	192	194	0*	192
4	12	18	583	590	587	588	499*	597
5	16	18	483	486	507	483	341*	489
6	14	18	610	610	622	610	574*	638
7	17	18	857*	857*	931	857*	857*	918
8	15	18	613	614	632	617	613*	639
9	16	18	1245	1245	1343	1245	919*	1295
10	15	18	328	331	407	332	149*	422
11	11	18	0*	0*	0*	12	0*	15
12	15	18	291	288	309	305	48*	307
13	15	18	1152	1156	1202	1152	854*	1161
14	14	18	32	33	34	38	20*	33
15	15	18	0*	0*	0*	0*	0*	3
16	8	18	0*	0*	0*	0*	0*	7
17	13	18	218	268	255	218	17*	270

*indica la mejor solución para cada problema

En la tabla II se resumen la cantidad de mejores y peores resultado para cada una de los procedimientos heurísticos. De los resultados mostrados se puede deducir que SA y GA han arrojado la mayoría de los peores resultados, y la metaheurística que ha logrado alcanzar los mejores resultados para todos los ejemplares estudiados es la Búsqueda Tabú.

Tabla II. Caso real. Cantidad de mejores y peores resultados.

Cantidad de soluciones:	peores	mejores*
AED	0	5
ANED	0	4
SA	5	3
GRASP	2	3
TS	0	17
GA	10	0

*(1 empate binario, 1 cuaternario, 2 quintuplos)

El 100% de las mejores soluciones se han obtenido con TS, seguido por la heurística AED con un 29% y ANED con un 24%. GA ha sido la metaheurística menos adecuada para resolver el problema estudiado, ya que en ninguno de los casos ha logrado alcanzar los mejores resultados. Ver figura 1.

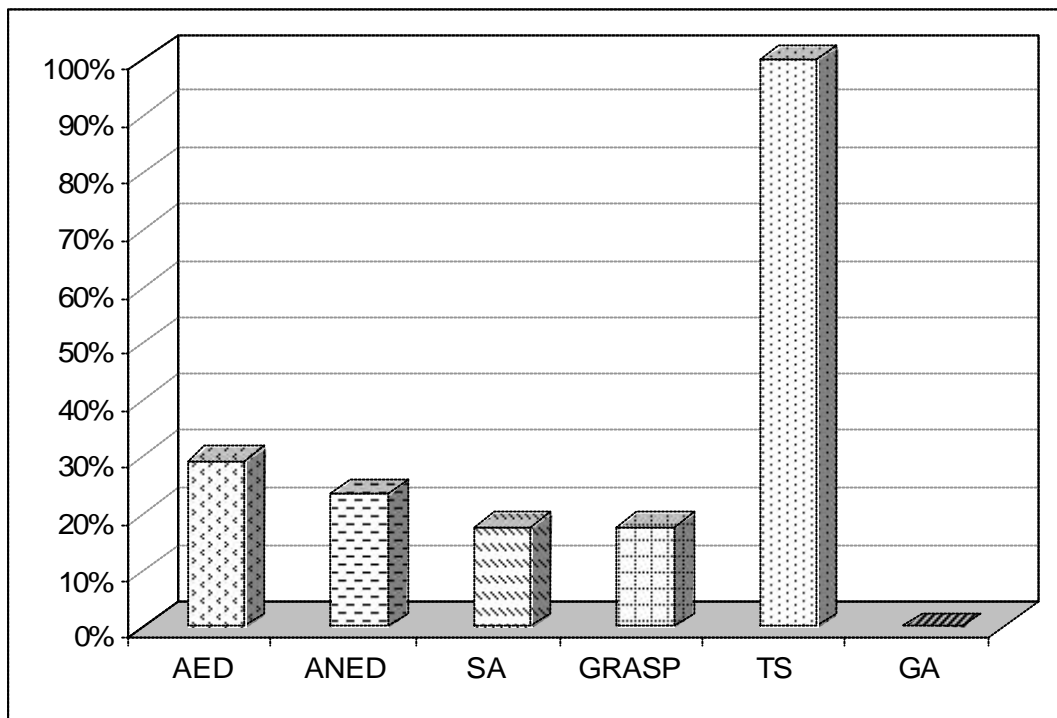


Figura 1. Caso real. Porcentaje de mejores resultados.

En la figura 2 se muestran los tiempos promedios de ejecución, expresados en segundos, necesarios para resolver un ejemplar usando cada uno de los procedimientos de exploración de entornos.

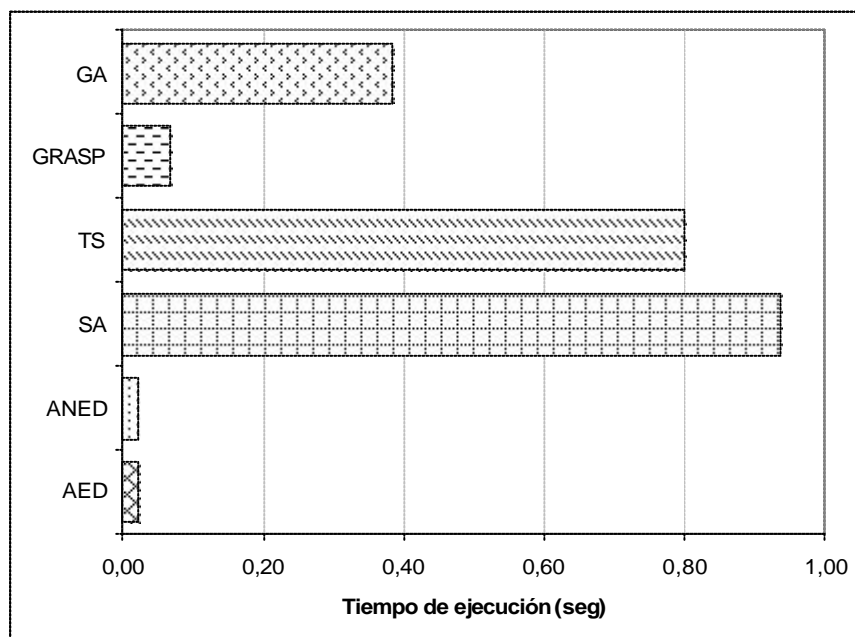


Figura 2. Caso real. Tiempo promedio de ejecución

De acuerdo con los resultados mostrados en la tabla II, los tiempos promedios para resolver un ejemplar no llegan a un (1) segundo, siendo las metaheurísticas SA, TS y GA las que consumen mayor cantidad de tiempo. Las heurísticas de optimización local AED y ANED son las que necesitan menor esfuerzo para obtener la solución.

Los porcentajes de error relativo de las soluciones obtenidas en cada uno de los ejemplares estudiados, se presentan en la tabla III.

De esta tabla se puede deducir que el procedimiento que arroja mejores resultados es la Búsqueda Tabú, ya que tiene un porcentaje de error relativo igual a cero en comparación con los demás procedimientos, lo que indica que todas las mejores soluciones se han obtenido con esta metaheurística. La baja dispersión de los resultados obtenidos con TS, demuestra que el procedimiento es estable. Además, se observa que para los otros procedimientos (AED, ANED, GRASP, SA y GA), el promedio del porcentaje de error relativo es muy elevado.

Tabla III. Caso real. Porcentaje de error relativo.

Ejemplar	PORCENTAJE DE ERROR RELATIVO					
	AED	ANED	SA	GRASP	TS	GA
1	1252,9	1423,5	1458,8	1482,4	0,0	1511,8
2	13,5	13,1	14,9	13,5	0,0	15,3
3	19200,0	19200,0	19200,0	19400,0	0,0	19200,0
4	16,8	18,2	17,6	17,8	0,0	19,6
5	41,6	42,5	48,7	41,6	0,0	43,4
6	6,3	6,3	8,4	6,3	0,0	11,1
7	0,0	0,0	8,6	0,0	0,0	7,1
8	0,0	0,2	3,1	0,7	0,0	4,2
9	35,5	35,5	46,1	35,5	0,0	40,9
10	120,1	122,1	173,2	122,8	0,0	183,2
11	0,0	0,0	0,0	1200,0	0,0	1500,0
12	506,3	500,0	543,8	535,4	0,0	539,6
13	34,9	35,4	40,7	34,9	0,0	35,9
14	60,0	65,0	70,0	90,0	0,0	65,0
15	0,0	0,0	0,0	0,0	0,0	300,0
16	0,0	0,0	0,0	0,0	0,0	700,0
17	1182,4	1476,5	1400,0	1182,4	0,0	1488,2

9. CONCLUSIONES

En este trabajo se ha tratado el problema de programación de operaciones en una máquina de la vida real con tiempos de preparación dependientes de la secuencia de los pedidos y familias de productos. El objetivo ha sido la minimización de la suma de los retrasos de los pedidos.

Se han utilizado cuatro estrategias para obtener una secuencia inicial, y se han aplicado seis procedimientos heurísticos de exploración de entornos: Algoritmos Exhaustivo de Descenso, Algoritmo No Exhaustivo de Descenso, Recocido Simulado, Búsqueda Tabú, GRASP y Algoritmos Genéticos.

Los resultados demuestran que la Búsqueda Tabú es un procedimiento que puede proporcionar buenas soluciones para el problema específico estudiado cuando se considera como objetivo el retraso total con tiempos de preparación dependientes de la secuencia, ya que necesita poco tiempo de procesamiento y logra las mejores soluciones. Las metaheurísticas Recocido Simulado y Algoritmo Genético son las que arrojan peores resultados, en vista de esto no es aconsejable su aplicación, para el ambiente industrial estudiado. Las Heurísticas de Optimización Local AED, ANED y la metaheurística GRASP pueden tener una ventaja sobre la Búsqueda Tabú para problemas con gran cantidad de pedidos, n , debido a que necesitan menos tiempo para encontrar una buena solución. Finalmente, es importante resaltar que estas conclusiones se han basado en algoritmos heurísticos específicos en un grupo específico de problemas.

La investigación presentada en este trabajo puede ser tomada como una base para posibles extensiones enfocadas a considerar otro tipo de variaciones del problema estudiado, incorporando elementos que en ocasiones también aparecen en determinados ambientes industriales reales. Por ejemplo: introducir restricciones de costos de penalización lo que permitirá considerar otro tipo de funciones objetivo como la minimización de la penalización total.

6. BIBLIOGRAFÍA

- Allahverdi A. y Aldowaisan T. (2000), "No-wait and separate setup three-machine flowshop with total completion time criterion", *International Transactions in Operational Research*, Vol. 7, No. 3, pp. 245-264
- Bulfin R. Jr. y Sipper D. (1998), "Planeación y Control de la Producción", Editorial Mc Graw Hill, México
- Companys R. y D'Armas M. (2005), "Programación de operaciones con tiempo de preparación mediante algoritmos de optimización local". *Universidad, Ciencia y Tecnología*, Vol. 9, No. 35, pp.155-162
- Conway RW, Maxwell WL y Miller LW. (1967). *Theory of Scheduling*. Addison Wesley, MA.
- De Castro R., Companys R. y Mateo M. (2003), "Programación de las órdenes de fabricación en una sola máquina con tiempos de preparación", *V Congreso de Ingeniería de Organización*, Valladolid-Burgos, pp. 59-60.
- Díaz A., Glover F., Ghasiri H., González JL., Laguna M., Moscazo P. y Tseng F. (1996), "Optimización Heurística y Redes Neuronales en Dirección de Operaciones e Ingeniería", Editorial Paraninfo, España
- Feo T. y Resende. (1989), "A probabilistic heuristic for a computationally difficult set covering problem", *Operations Research Letters*. Vol. 8, pp.67-71
- Fernández-Baños I. (2003), *Programación de la secuencia de fabricación en una máquina, con tiempos de preparación variables, mediante la aplicación de Algoritmos Genéticos*. Proyecto de Fin de Carrera. Escuela Técnica Superior de Ingeniería Industrial de Barcelona, España
- Glover F. (1989), "Tabu Search, Part I", *ORSA Journal on Computing*, Vol. 1, pp.190-206.
- Glover F. y Melián B. (2003), "Búsqueda Tabú", *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, Vol. 19, pp. 29-48
- Holland J. (1975), *Adaptation in Natural and Artificial Systems*, Ann Arbor: University of Michigan Press.
- Johnson D., Aragon C., McGeoch L. y Sherron C. (1989). *Optimization by Simulated Annealing: An experimental Evaluation; Part I. Graph Partitioning*. *Operation Research*. Vol. 37, issue 6, p. 865-892
- Kirkpatrick S., Gelatt CD. y Vecchi MP. (1983), "Optimization by simulated annealing", *Science*, Vol. 220, pp. 671-680.
- Raghu T.S. y Rajendran C. (1995), "Due-date setting methodologies based on simulated annealing-an experimental study in a real-life job shop", *International Journal of Production Research*, Vol. 33, No. 9, pp. 2535-2554

- Rajendran C. y Ziegler H. (2003), "Scheduling to minimize the sum of weighted flowtime and weighted tardiness of jobs in a flowshop with sequence-dependent setup times", *European Journal of Operational Research*, Vol. 149, pp. 513-522
- Rivas I. y Companys R. (2004), "Secuenciación de pedidos en una máquina con tiempos de preparación variables". Documento Interno de Trabajo. Laboratorio de Organización Industrial. Departamento de Organización de Empresas. Universidad Politécnica de Cataluña. España
- Yang Y., Kreipl S. y Pinedo M. (2000), "Heuristics for minimizing total weighted tardiness in flexible flow shops", *Journal of Scheduling*, Vol. 3, pp.89-108